AN ALTERNATIVE APPROACH FOR CONSTRUCTIVE ENUMERATION OF GRAPHS

Jiří POSPÍCHAL and Vladimír KVASNIČKA

Department of Mathematics,
Slovak Technical University, 81237 Bratislava, The Slovak Republic

Received November 27, 1991
Accepted March 16, 1992

An efficient method for constructive enumeration of graphs is suggested. The method is based on the so-called semicanonical numbering of graphs, that is a numbering much more restrictive than the cooperative numbering. Graph-theoretical properties of the semicanonical numbering make it possible to formulate an exhaustive and nonredundant constructive enumeration of graphs. The approach allows an easy introduction of specifications for molecular graphs.

The problem of constructive enumeration of molecular graphs has been successfully solved only for acyclic molecules[1 - 4]. The constructive enumeration of cyclic molecular graphs is usually solved[5 - 14] by making use of an enormous number of isomorphism checks. Some additional conditions are introduced and used to reduce a priori the number of the produced graphs – candidates to be verified in the subsequent isomorphism checks[15]. To this purpose many beautiful group-theoretical and combinatorial concepts have been elaborated[16 - 18].

Recently, we have suggested[19 - 20] a graph-theoretical method, suitable for the constructive enumeration of molecular graphs, which employs the canonical numbering based on the maximum code produced by the lower triangle part of adjacency matrix. Almost simultaneously a very similar approach (employing the maximum code produced by the upper-triangle part of adjacency matrix) has been published by Hendrickson and Parks[21]. Both these approaches use the concept of the so-called *cooperative numbering* of graphs[21 - 23] initially suggested by Morgan[22]. It substantially reduced the total number of all possible numberings (n! problem) to a much smaller, more manageable number.

Fifteen years ago a series of basic Faradzhev's published papers[24 - 27] (see also ref.[28]) was devoted to the constructive enumeration of combinatorial objects. The papers hardly received any attention, being rather difficult to read and partly published in a hardly accessible journal. The principal ideas of this method were described by a combination of rather abstract mathematical tools and highly sophisticated algorithms, without giving any examples.

In the present paper Faradzhev's concept of semicanonical adjacency matrix is reconstituted, even though ideas leading to theorems 2 – 4 could be derived from Faradzhev's works. Basic improvements of this approach in the present paper consist in (i) a reformulation of Faradzhev's method by making use simple but sufficiently precise algebraical and graph-theoretical tools and (ii) in an introduction of further conditions for canonicity of the so-called semicanonical matrices used in constructive enumeration of graphs.

The semicanonical numbering is much more restrictive than the cooperative numbering of graphs, which we have used earlier. Although a semicanonically numbered graph is also cooperative numbered, the reverse statement, in general, is not true. Moreover, there exists a simple proof that each canonically numbered graph (according to the maximum code of the upper-triangle part of adjacency matrix) must also be semicanonically numbered. Therefore, when looking for the canonically numbered graphs it is entirely sufficient to apply only those numberings that are a priori semicanonical. This property allows the formulation of an efficient recurrent method for the constructive enumeration of graphs. To our knowledge and experience, the constructive enumeration employing the notion of semicanonical numbering offers the most efficient algorithm known.

THEORETICAL

Basic Concepts

The purpose of this section is to present a concept of the so-called *canonical adjacency matrix*, which belongs to the principal notions of Faradzhev's approach[24] to constructive enumeration of graphs.

Let $\mathcal{F}_{pq}$ (for $p > 0$ and $q \geq 0$) be a *family of adjacency matrices* $\boldsymbol{A} = (a_{ij})$ of simple graphs with p vertices and q edges[29]. Vertices will be labeled by the first p natural numbers.

A permutation $R = (r_1, r_2, \ldots, r_p)$ of p objects $(1, 2, \ldots, p)$ is uniquely represented[30] by the so-called permutation matrix $\boldsymbol{R}$ composed of zero and unit entries, where each row/column contains just one unit entry. A set (called *symmetry group*) of all permutation matrices of p objects will be denoted by $\mathcal{S}_p$.

Definition 1. Two adjacency matrices $\boldsymbol{A}_1, \boldsymbol{A}_2 \in \mathcal{F}_{pq}$ are called *equivalent*, $\boldsymbol{A}_1 \cong \boldsymbol{A}_2$, if there exists a permutation matrix $\boldsymbol{R} \in \mathcal{S}_p$ such that

$$\boldsymbol{A}_1 = \boldsymbol{R}^T \boldsymbol{A}_2 \boldsymbol{R}. \qquad (1)$$

Two equivalent matrices $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$, $\boldsymbol{A}_1 \cong \boldsymbol{A}_2$, are represented by graphs $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$, respectively, that are *isomorphic*, $\boldsymbol{G}_1 \cong \boldsymbol{G}_2$, formally $\boldsymbol{A}_1 \cong \boldsymbol{A}_2 \Leftrightarrow \boldsymbol{G}_1 \cong \boldsymbol{G}_2$.

According to the above introduced relation of equivalence between adjacency matrices, the family $\mathcal{F}_{pq}$ may be decomposed onto disjoint subfamilies that are composed of equivalent matrices,

$$\mathcal{F}_{pq} = \mathcal{F}_{pq}^{(1)} \cup \mathcal{F}_{pq}^{(2)} \cup \ldots , \tag{2}$$

where $\boldsymbol{A}_1, \boldsymbol{A}_2 \in \mathcal{F}_{pq}^{(i)} \Leftrightarrow \boldsymbol{A}_1 \cong \boldsymbol{A}_2$.

We assign to each adjacency matrix $\boldsymbol{A} \in \mathcal{F}_{pq}$ a *string* composed of its upper-triangle entries[23,32],

$$[\boldsymbol{A}] = (a_{12} \ldots a_{1p}\, a_{23} \ldots a_{2p} \ldots a_{p-1,p}) . \tag{3}$$

These strings determine unambiguously the corresponding adjacency matrix. The strings are related by the following three lexicographical relations: "equal to", "smaller than", and "greater than", i.e. $[\boldsymbol{A}_1] = [\boldsymbol{A}_2]$, $[\boldsymbol{A}_1] < [\boldsymbol{A}_2]$, and $[\boldsymbol{A}_1] > [\boldsymbol{A}_2]$, respectively. There exist also other possibilities how to build such a strings[33]. The present one has not any chemical meaning, it was chosen only for its useful properties appropriate for constructive enumeration of graphs.

Each subfamily $\mathcal{F}_{pq}^{(i)}$ is represented by its *code*. We introduce two types of codes, corresponding to the maximum or minimum value of strings of matrices from the subfamily $\mathcal{F}_{pq}^{(i)}$,

$$\mathrm{CODE}_i = \max_{\boldsymbol{A} \in \mathcal{F}_{pq}^{(i)}} [\boldsymbol{A}] , \tag{4a}$$

$$\mathrm{code}_i = \min_{\boldsymbol{A} \in \mathcal{F}_{pq}^{(i)}} [\boldsymbol{A}] . \tag{4b}$$

The entity CODE_i, determined as the maximum value of strings of adjacency matrices from the subfamily $\mathcal{F}_{pq}^{(i)}$, was initially introduced by Hendrickson et al.[23] for purposes of chemical informatics. Recently, Hendrickson et al.[21] have used this code also for the constructive enumeration of graphs. The second alternative (dual) approach, based on the minimum value of their strings, has been introduced and extensively applied by Randic[32]. In our forthcoming considerations we show (see Theorem 1) that these two different approaches are closely related.

The definitions (*4a*) – (*4b*) may be alternatively rewritten as follows:

$$\mathrm{CODE}_i = \max_{\boldsymbol{R} \in S_p} [\boldsymbol{R}^T \boldsymbol{A}\, \boldsymbol{R}] , \tag{5a}$$

$$\text{code}_i = \min_{\boldsymbol{R} \in \mathcal{S}_p} [\boldsymbol{R}^T \boldsymbol{A} \boldsymbol{R}], \tag{5b}$$

where $\boldsymbol{A} \in \mathcal{F}_{pq}^{(i)}$.

Definition 2. Adjacency matrices $\hat{\boldsymbol{A}}$, $\tilde{\boldsymbol{A}} \in \mathcal{F}_{pq}^{(i)}$ with the codes determined by

$$\hat{\boldsymbol{A}} : \text{CODE}_i = [\hat{\boldsymbol{A}}] = \text{CODE}(\hat{\boldsymbol{A}}), \tag{6a}$$

$$\tilde{\boldsymbol{A}} : \text{code}_i = [\tilde{\boldsymbol{A}}] = \text{code}(\tilde{\boldsymbol{A}}), \tag{6b}$$

are called the *canonical adjacency matrices.*

Of course, there exist many other ways how to determine this concept. The present way of determination of canonical adjacency matrices has been mainly determined by its useful properties for our constructive enumeration of graphs. We have to emphasize that the term "canonical" should be distinguished for two separate cases. It is necessary to specify whether the "maximum" or the "minimum" code is used.

The canonical matrix $\hat{\boldsymbol{A}}$ (or $\tilde{\boldsymbol{A}}$) is a representative of the subfamily $\mathcal{F}_{pq}^{(i)}$. A graph $\hat{G}$ ($\tilde{G}$) assigned to the canonical adjacency matrix $\hat{\boldsymbol{A}}$ ($\tilde{\boldsymbol{A}}$) will be called the *canonically numbered graph.*

Let $\overline{\boldsymbol{A}} = (\overline{a}_{ij})$ be a matrix assigned to an adjacency matrix $\boldsymbol{A} \in \mathcal{F}_{pq}$ as a complement determined by

$$\overline{a}_{ij} = \begin{cases} 0 & (\text{if } i = j), \\ 1 & (\text{if } i \neq j \text{ and } a_{ij} = 0), \\ 0 & (\text{if } i \neq j \text{ and } a_{ij} = 1). \end{cases} \tag{7}$$

This means, that the matrix $\overline{\boldsymbol{A}}$ is also an adjacency matrix and it belongs to a family $\mathcal{F}_{p\overline{q}}$, where $\overline{q} = (p(p-1)/2) - q$. The adjacency matrix $\overline{\boldsymbol{A}}$ is called the *complement* of $\boldsymbol{A}$. In a similar way we introduce a complement of a string $[\boldsymbol{A}]$, formally $\rceil [\boldsymbol{A}] = [\overline{\boldsymbol{A}}]$, e.g. $\rceil (111100) = (000011)$. A graph $\overline{G}$ assigned to the adjacency matrix $\overline{\boldsymbol{A}}$ is nothing else than a *complementary graph* G assigned to the graph G.

Theorem 1. The codes $\text{CODE}(\boldsymbol{A})$ and $\text{code}(\boldsymbol{A})$ are related by

$$\text{code}(\boldsymbol{A}) = \rceil \text{CODE}(\overline{\boldsymbol{A}}). \tag{8}$$

According to this theorem $\text{code}(\boldsymbol{A})$, determined as the minimum value of strings, see (*4b*) or (*5b*), may be simply constructed in such a way that we first construct a "maximum" code of the matrix $\overline{\boldsymbol{A}}$. Then a complement of the produced string determines

code($\boldsymbol{A}$). The proof of this theorem immediately follows from the relations (*4a*) and (*4b*).

It may seem, at first sight, that the application of both codes CODE($\boldsymbol{A}$) and code($\boldsymbol{A}$) is fully equivalent from the computational point of view. Unfortunately, such conclusion is incorrect for the following reasons. An efficient algorithm for finding a canonical numbering of graphs will strongly depend on the number of vertices that may be potentially numbered by 1. According to the CODE($\boldsymbol{A}$) these vertices should be of the greatest valence[19] and for code($\boldsymbol{A}$) of the smallest valence. For graphs with "small density" of edges a number of vertices of greatest valence is usually considerably smaller than the number of vertices of smallest valence. This means that graphs with small number of edges (i.e. $q << p(p-1)/2$) are easily numbered by the canonical numbering according to the "maximum" CODE($\boldsymbol{A}$). After all, a numerical construction of "maximum" code for molecular graphs is much more efficient than its dual "minimum" counterpart initially introduced by Randic[32].

For a fixed s bounded by $1 \le s < p$ an adjacency matrix $\boldsymbol{A} \in \mathcal{T}_{pq}$ may be decomposed onto block matrices as follows:

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{12}^{T} & \boldsymbol{A}_{22} \end{pmatrix}, \tag{9}$$

where $\boldsymbol{A}_{11}$ ($\boldsymbol{A}_{22}$) corresponds to the left-up (right-bottom) corner submatrix of $\boldsymbol{A}$, its type is $t(\boldsymbol{A}_{11}) = (s,s)$ [$t(\boldsymbol{A}_{22}) = (p-s, p-s)$]. A rectangular submatrix $\boldsymbol{A}_{12}$ of the type $t(\boldsymbol{A}_{12}) = (s, p-s)$ will be expressed via its s-dimensional column vectors,

$$\boldsymbol{A}_{12} = (\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_{p-s}) . \tag{10}$$

Let $\boldsymbol{a} = (a_i)$ and $\boldsymbol{b} = (b_i)$ be two s-dimensional column vectors, they are equal, $\boldsymbol{a} = \boldsymbol{b}$, if $a_i = b_i$, for $i = 1, 2, \ldots, s$. These vectors may be also related by $\boldsymbol{a} > \boldsymbol{b}$ ($\boldsymbol{a} < \boldsymbol{b}$) if there exists such an integer $1 \le i \le s$ that $a_j = b_j$, for $j = 1, 2, \ldots, i-1$, and $a_i > b_i$ ($a_i < b_i$).

Definition 3. An adjacency matrix $\boldsymbol{A} \in \mathcal{T}_{pq}$ is called *nonincreasingly semicanonical* if for each $1 \le s < p$ the column vectors $\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_{p-s}$ of the submatrix $\boldsymbol{A}_{12}$ satisfy

$$\boldsymbol{c}_1 \ge \boldsymbol{c}_2 \ge \ldots \ge \boldsymbol{c}_{p-s} . \tag{11a}$$

The adjacency matrix $\boldsymbol{A} \in \mathcal{T}_{pq}$ is called *nondecreasingly semicanonical* if for each $1 \le s < p$ the column vectors satisfy

$$\boldsymbol{c}_1 \le \boldsymbol{c}_2 \le \ldots \le \boldsymbol{c}_{p-s} . \tag{11b}$$

In order to keep our discussion as simple as possible we shall use, instead of the clumsy terms "nonincreasingly semicanonical", and "nondecreasingly semicanonical" only the shorter term "semicanonical". The type of semicanonicity and also the type of canonicity under consideration will be always specified in advance. A graph determined by a semicanonical adjacency matrix will be called the *semicanonically numbered graph*.

Theorem 2. A canonical adjacency matrix is semicanonical.

Let us consider an adjacency matrix $\boldsymbol{A} \in \mathcal{F}_{pq}$ and let us assume that for a fixed $1 \leq s < p$, the vectors $c_1, c_2, \ldots, c_{p-s}$ of the rectangular matrix $\boldsymbol{A}_{12}$ do not satisfy the condition (*11a*) [or the condition (*11b*) when the "minimum" code (*4b*) is used]. For instance, the column vectors c_i and c_j (for $1 \leq i < j \leq s$) satisfy $c_i < c_j$, then there should exist a permutation matrix $\boldsymbol{R}$, which corresponds to a transposition of indices i and j such that $[\boldsymbol{R}^T\boldsymbol{A}\boldsymbol{R}] > [\boldsymbol{A}]$. In fact, we have proved that an adjacency matrix which is not semicanonical cannot be canonical; a reverse form of this implication gives the above theorem.

Theorem 2 represents an important necessary condition of the canonicity of adjacency matrices. If we are looking for a canonical matrix in the subfamily $\mathcal{F}_{pq}^{(i)}$, then it is sufficient to consider only those matrices of $\mathcal{F}_{pq}^{(i)}$ that are semicanonical, all other adjacency matrices can be skipped.

Now we turn our attention to the possibility of formulation of another type of necessary conditions for canonicity of adjacency matrices. Let for some fixed $1 \leq s < p$ the adjacency matrix (*9*) be rewritten in the so-called *s-reduced adjacency matrix*,

$$\boldsymbol{A}_s = \begin{pmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{12}^T & \boldsymbol{O} \end{pmatrix}. \tag{12}$$

The matrix $\boldsymbol{A}_s$ is formed from the adjacency matrix $\boldsymbol{A}$ in such a way that the right-bottom block $\boldsymbol{A}_{22}$ is substituted by a null matrix. The matrix $\boldsymbol{A}_s$ is also an adjacency matrix and belongs to a family $\mathcal{F}_{pq'}$, where $q' \leq q$. If the adjacency matrix $\boldsymbol{A}$ corresponds to a graph G, then the matrix $\boldsymbol{A}_s$ corresponds to a subgraph G_s formed from G by deleting all edges $[i, j] \in E(G)$, for $i, j \in \{s + 1, s + 2, \ldots, p\}$.

Theorem 3. If an adjacency matrix $\boldsymbol{A} \in \mathcal{F}_{pq}$ is canonical, then for each $1 \leq s < p$ the s-reduced adjacency matrices $\boldsymbol{A}_s$ are also canonical.

Its proof may be done in a similar fashion as the proof of theorem 2.

Canonical Numbering of Graphs

In this section the term "canonical" will be related to the approach based on the "maximum" code of adjacency matrices. The concept of canonical numbering of a graph G, represented by an adjacency matrix $\boldsymbol{A} \in \mathcal{J}_{pq}^{(i)}$ (i.e. the graph G is composed of p vertices and q edges) consists in a finding a permutation matrix $\boldsymbol{R} \in \mathcal{S}_p$ such that (see (*5a*))

$$\mathrm{CODE}_i = [\boldsymbol{R}^T \boldsymbol{A} \boldsymbol{R}] . \qquad (13)$$

If the graph G is canonically numbered, then the above relation is achieved automatically for an identical permutation, $\boldsymbol{R} = \boldsymbol{E}$, all other permutation matrices that satisfy (*13*) correspond to *automorphisms* of the graph. For some special cases of interest (e.g. for our constructive enumeration of graphs) we need only to know whether a graph is canonically numbered or not; if we have found a matrix $\boldsymbol{R}$ which gives a string satisfying $[\boldsymbol{R}^T\boldsymbol{A}\boldsymbol{R}] > [\boldsymbol{A}]$, then the graph G is not canonically numbered.

The main effort in finding canonical numberings of graphs should be concentrated on a process of achieving (*13*) by making use of only those permutation matrices $\boldsymbol{R} \in \mathcal{S}_p$ that lead as fast as possible to the required result. A vertex of G which will be in this process numbered by 1 should be of the highest valence, all other vertices will give adjacency matrices with the first row smaller than that one of the matrix $\hat{\boldsymbol{A}}$ corresponding to the maximum code $\mathrm{CODE}(\hat{\boldsymbol{A}})$, see (*6a*). This process is considerably accelerated by theorem 3, a canonical numbering may be successively constructed by the trial and error method (implemented as a *backtrack searching* algorithm with a *branch and bound* modification[32]) in such a way that the created reduced adjacency matrices are canonical. Furthermore, only those permutation matrices $\boldsymbol{R} \in \mathcal{S}_p$ may be used (see theorem 2) that will produce semicanonical adjacency matrices. Combining these two observations we get the very efficient method of canonical numbering of graphs which, in a process of traversing through searching tree, skips all permutations that have no chance to produce the canonical numbering.

Semicanonical Numbering

Let G be a connected graph and let $v_0 \in V(G)$ be an arbitrary vertex (its selection will be specified in such a way that it can potentially produce maximum code), then the vertex set $V(G)$ may be decomposed onto disjoint subsets as follows:

$$\mathcal{D}(v_0) : V(G) = V_1 \cup V_2 \cup \ldots \cup V_i \ldots , \qquad (14a)$$

$$V_i = \{v \in V(G);\ d(v, v_0) = i - 1\}, \ (\text{for } i = 1, 2, \ldots) , \qquad (14b)$$

where $V_1 = \{v_0\}$ and $d(v, v_0)$ is the distance between vertices v and v_0 (i.e. the length of a shortest path connecting the vertices v and v_0). The subset $V_i \subseteq V(G)$ may be interpreted as the $(i - 1)$-th sphere of the vertex v_0, the distance between this vertex and a vertex of V_i is determined by $d(v, v_0) = i - 1$. The decomposition $\mathcal{D}(v_0)$ determined by (*14*) may be schematically visualized by a graph with vertically situated subsets of vertices. Going from the top to the bottom of graph the subset V_1 comes the first, then come subsets V_2, V_3, ... A pair of vertices can be adjacent if and only if both belong to the same subset (i.e. from the same level) or they belong to different subsets lying on juxtaposed levels.

It is easy to see that adjacency matrices assigned to graphs with vertex sets decomposed in a manner specified by (*14*) may be written as follows.

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} & \boldsymbol{O} & \boldsymbol{O} & . & . & . \\ \boldsymbol{A}_{12}^{\mathrm{T}} & \boldsymbol{A}_{22} & \boldsymbol{A}_{23} & \boldsymbol{O} & . & . & . \\ \boldsymbol{O} & \boldsymbol{A}_{23}^{\mathrm{T}} & \boldsymbol{A}_{33} & \boldsymbol{A}_{34} & . & . & . \\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{A}_{34}^{\mathrm{T}} & \boldsymbol{A}_{44} & . & . & . \\ . & . & . & . & . & . & . \end{pmatrix} \qquad (15)$$

The diagonal block matrix $\boldsymbol{A}_{ii}$ corresponds to a subgraph $G_{ii} \subseteq G$ induced by a vertex subset V_i whereas the nondiagonal block matrix $\boldsymbol{A}_{i,\,i+1}$ corresponds to a bipartite subgraph $G_{ij} \subseteq G$ induced by V_i and V_{i+1}, where only those edges are considered that are simultaneously incident with a vertex of V_i and a vertex of V_{i+1}. Since there do not exist edges connecting two vertices with distance larger than 1, all other nondiagonal block matrices $\boldsymbol{A}_{ij}$, for $j > i + 1$, are null matrices.

A semicanonicity of an adjacency matrix, expressed in the form (*15*), is dominated by semicanonicities of nondiagonal blocks $\boldsymbol{A}_{12}$, $\boldsymbol{A}_{23}$, $\boldsymbol{A}_{34}$, ... Only if for a block $\boldsymbol{A}_{i,\,i+1}$ some column vectors happen to be mutually equal, a semicanonicity of its diagonal counterpart $\boldsymbol{A}_{i+1,\,i+1}$ may play a dominant role.

For the decomposition (*15*) we assign to each subset V_i an index subset U_i in such a way that by going successively from V_1 via V_2, V_3 , . . ., an actual index set U_i is filled-up by successive positive integers

$$\begin{aligned} \mathcal{U}[\mathcal{D}(v_0)] : \{1, 2, \ldots, p\} &= U_1 \cup U_2 \cup U_3 \cup \ldots, \\ U_1 &= \{1\}, \\ U_2 &= \{2, 3, \ldots, |V_1| + 1\}, \\ U_3 &= \{|V_1| + 2, |V_1| + 3, \ldots, |V_1| + |V_2| + 1\}, \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \end{aligned} \qquad (16)$$

Let us assume that for a graph G (represented by an adjacency matrix $\boldsymbol{A} \in \mathcal{T}_{pq}$) and a vertex $v_0 \in V(G)$ we know the decomposition $\mathcal{D}(v_0)$ and the index system $\mathcal{U}[\mathcal{D}(v_0)]$. A semicanonical numbering of the graph G consist of a construction of the 1–1 maping,

$$R : V(G) = \{1, 2, \ldots, p\} \rightarrow \{1, 2, \ldots, p\} . \tag{17a}$$

Its construction will be successively carried out in a form of some restricted 1–1 mappings (their actual forms will be specified later in the text)

$$R : V_i \rightarrow U_i , \tag{17b}$$

for $i = 1, 2, \ldots$. This mapping assigns unambiguously to each vertex from V_i an index from U_i. The mapping may be simply interpreted as a permutation $R = (r_1, r_2, \ldots, r_p)$ of p objects $(1, 2, \ldots, p)$; entry r_i determines an assigned number to the i-th vertex from $V(G)$.

The concept of semicanonically numbered graphs has been initially introduced (see comment above theorem 2) through its adjacency matrix which has to be also semicanonical. This definition of semicanonicity is not constructive, by applying it we can only check whether a graph is semicanonically numbered or not. It could not be used for a construction of semicanonical numbering of graphs. For our constructive enumeration of graphs it is vitally important to have a handy method which is able to construct the semicanonical numbering of graphs irrespective of their initial numbering.

Let us assume that an arbitrary vertex $v_0 \in V(G)$ has been chosen as a vertex which is numbered by 1, formally $r_1 = v_0$. The vertex set $V(G)$ will be divided into two disjoint subsets Y_1 and Y_2, where the first (second) one is composed of all vertices adjacent (nonadjacent) with the vertex v_0,

$$Y_1 = \Gamma(r_1), \tag{18a}$$

$$Y_2 = (V(G) \setminus \{v_0\}) \setminus Y_1 . \tag{18b}$$

The term $\Gamma(r_1)$ corresponds to a subset of $V(G)$ composed of all vertices that are adjacent with the vertex v_0. The subset Y_1 is equal to the subset V_2 of the decomposition (*14*) carried out with respect to the chosen vertex v_0, i.e. it determines the first sphere of v_0 in the graph G whereas the subset Y_2 is a union of all remaining subsets $V_3, V_4, \ldots, Y_2 = V_3 \cup V_4 \cup \ldots$. The next vertex which will be numbered by 2 is taken from the subset Y_1.

Let us assume that $(i - 1)$ vertices have been already numbered by $1, 2, \ldots, i - 1$ and that we have available the following system of sets $Y_1, Y_2, \ldots, Y_t$. An arbitrary vertex $v \in Y_1$ is numbered by i, $r_i = v$ and we create auxiliary sets

$$Y_k' = \Gamma(r_i) \cap Y_k , \tag{19a}$$

$$Y_k'' = \begin{cases} (Y_k \setminus \{v\}) \setminus Y_k' & (\text{for } k = 1) \\ Y_k \setminus Y_k' & (\text{for } k = 2, \ldots, t) \end{cases} \tag{19b}$$

for $k = 1, 2, \ldots, t$. That is, each subset Y_k has been divided into two disjoint subsets Y_k', Y_k'', where the first (second) one is composed of those vertices of Y_k that are (not) adjacent with the chosen vertex v. A sequence of auxiliary subsets Y_1', Y_1'', Y_2', Y_2'', . . ., Y_t', Y_t'' is compressed in a new sequence in such a way that only nonempty subsets are taken into account; we get Y_1, Y_2, . . ., $Y_{t'}$, where t' determines a number of nonempty subsets. This process is repeated until all vertices are numbered. The sets Y_1, Y_2, . . ., Y_t have the following simple interpretation: A set Y_i (for $i = 1, 2, \ldots, t$) is composed of those still nonnumbered vertices that are similar[20] from the viewpoint of already numbered vertices, when edges between nonnumbered vertices are not considered, see Fig. 1.

A verification of the above recurrent method is carried out simply by tracing what its single steps have done. In particular, the subsets Y_1, Y_2, . . ., Y_t are constructed in such a way that analogs of the condition (*11a*) are satisfied.

The described constructive method of semicanonical numbering of graphs is summarized in a form of recurrent algorithm implemented in a pseudo-PASCAL form.

Algorithm 1.

```
1     Y1 : = V(G); i : = 0; t : = 1;
2     WHILE t > 0 DO
4     BEGIN i : = i + 1; v : = an arbitrary vertex of Y1;
5           ri : = v;
            Y1 : = Y1 \ {v};
6           FOR k = 1 TO t DO
7           BEGIN Yk' : = Γ(ri) ∩ Yk; Yk'' : = Yk \Yk' END;
8           t' : = 0;
9           FOR k : = 1 TO t DO
10          BEGIN IF Yk' ≠ ∅ THEN BEGIN t' : = t' + 1; Yt' : = Yk' END;
11                IF Yk'' ≠ ∅ THEN BEGIN t' : = t' + 1; Yt' : = Yk'' END;
12          END;
13          t : = t';
14    END;
```

Since the concept of semicanonical numbering of graphs is given in a relatively abstract way, the following illustrative example is highly recommended for better understanding.

Example 1. The method of semicanonical numbering of graphs is illustrated on an example of graph A displayed in Fig. 1. In the initialization step the role of a "reference" vertex is played by the vertex denoted by C. Decomposition $\mathcal{D}(C)$ and an index system $\mathcal{U}[\mathcal{D}(C)]$ are

$$\mathcal{D}(C) : V_1 = \{C\},\ V_2 = \{B, D\},\ V_3 = \{A, E, F, G\},$$

$$\mathcal{U}[\mathcal{D}(C)] : U_1 = \{1\},\ U_2 = \{2, 3\},\ U_3 = \{4, 5, 6, 7\}.$$

Single steps of the algorithm of semicanonical numbering for the given graph look as follows:

Step 0. (Initialization.) $Y_1 := V(G) = \{A, B, \ldots, G\}$, $i = 0$.
Step 1. $v = C$, $i = 1$, $r_1 = C$, $Y_1 = \{B, D\}$, $Y_2 = \{A, E, F, G\}$.
Step 2. $v = D$, $i = 2$, $r_2 = D$, $Y_1 = \{B\}$, $Y_2 = \{E, G\}$, $Y_3 = \{A, F\}$.
Step 3. $v = B$, $i = 3$, $r_3 = B$, $Y_1 = \{G\}$, $Y_2 = \{E\}$, $Y_3 = \{A, F\}$.
Step 4. $v = G$, $i = 4$, $r_4 = G$, $Y_1 = \{E\}$, $Y_2 = \{F\}$, $Y_3 = \{A\}$.
Step 5. $v = E$, $i = 5$, $r_5 = E$, $Y_1 = \{F\}$, $Y_2 = \{A\}$.
Step 6. $v = F$, $i = 6$, $r_6 = F$, $Y_1 = \{A\}$.
Step 7. $v = A$, $i = 7$, $r_7 = A$, (algorithm is finished, the achieved semicanonical numbering is determined by the permutation $R = (r_1, r_2, \ldots, r_7)$).

The step 1 was carried out over the first level V_1, whereas steps 2 to 3 and steps 4 to 7 were carried out over the second and third level, respectively. The constructed semicanonical numbering is shown in Fig. 1 (step 7), the corresponding adjacency matrix (its upper triangle part) is

$$\boldsymbol{A} = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ & 0 & 0 & 1 & 1 & 0 & 0 \\ & & 0 & 1 & 0 & 1 & 1 \\ & & & 0 & 0 & 1 & 0 \\ & & & & 0 & 0 & 0 \\ & & & & & 0 & 1 \\ & & & & & & 0 \end{pmatrix} .$$

We see that this adjacency matrix is obviously semicanonical (but not canonical, for this case a vertex numbered by 1 should be selected B). The column vectors of nondia-

gonal block $\boldsymbol{A}_{23}$ are nonicreasing, when going from its left-hand side. Similarly, the diagonal blocks $\boldsymbol{A}_{22}$ and $\boldsymbol{A}_{33}$ are also separately semicanonical.

Canonical Numbering

The method of semicanonical numbering of graphs outlined in the previous subsection can now be deployed to suggest an efficient approach for the canonical numbering. The approach will be formulated in a form suitable for the constructive enumeration of graphs; it will give only an answer to the question whether an *s*-reduced adjacency matrix is canonical or not.

Let us consider an *s*-reduced adjacency matrix $\boldsymbol{A} \in \mathcal{J}_{pq}$ of the form (*12*). We look for a mapping/permutation $R = (r_1, r_2, \ldots, r_p)$ that would produce a maximum code. To do this it suffices to consider only those permutations that give, a priori, the semicanonical adjacency matrices. This means that algorithm 1 may be simply modified in such a way that it will produce an algorithm for checking the canonicity of an *s*-reduced adjacency matrix.

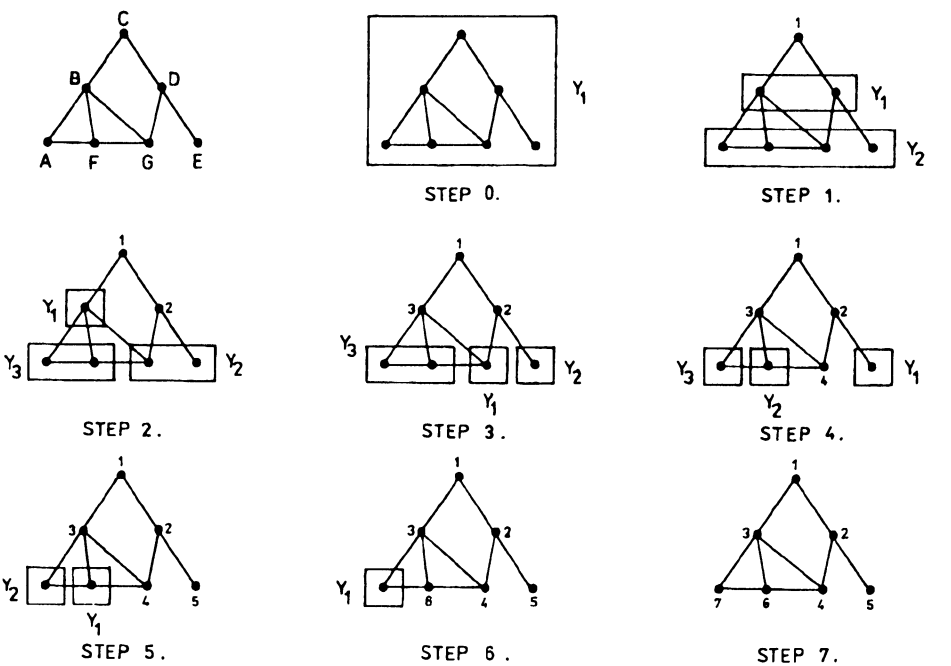

FIG. 1
An example of semicanonical numbering

Algorithm 2.

```
1   Y1^(1) := {1, 2, ..., p}; U1 := {k; val(k) = val_max};
    i := 1; t1 := 1; canonicity := true;
2   REPEAT IF U1 ≠ ∅ THEN
3   BEGIN r_i := min(U_i); U_i := U_i \ {r_i}; Y1^(i) := Y1^(i) \ {r_i};
4         FOR k := 1 TO t_i DO
5         BEGIN Y_k' := Γ(r_i) ∩ Y_k^(i); Y_k'' := Y_k^(i) \ Y_k'; END;
6         Y1^(i) := Y1^(i) ∪ {r_i}; j := 0;
7         FOR k := 1 TO t_i DO
8         BEGIN IF Y_k' ≠ ∅ THEN BEGIN j := j + 1; Y_j^(i+1) := Y_k' END;
9               IF Y_k'' ≠ ∅ THEN BEGIN j := j + 1; Y_j^(i+1) := Y_k'' END
10        END; t_(i+1) := j; l := i;
11        FOR k := 1 TO t_(i+1) DO FOR j := 1 TO p DO
12        IF j ∈ Y_k^(i+1) THEN BEGIN l := l + 1; r_l := j END;
13        IF row_i = ROW_i THEN
14        BEGIN IF i = s THEN
15              BEGIN i := min {k; r_k ≠ k};
16                    FOR j := 1 TO i - 1 DO FOR k := 1 TO s DO
17                    IF r_k < k THEN U_j := U_j \ {k};
18              END ELSE BEGIN i := i + 1; U_i := Y1^(i) END
19        END ELSE IF row_i > ROW_i THEN canonicity := false
20  END ELSE i := i - 1;
21  UNTIL (i = 0) or (not canonicity);
```

The value of boolean variable 'canonicity' is true (false) if the adjacency matrix is canonical (not canonical). The set U_1 (see row 1) is composed of indices that correspond to vertices of maximum valence. The symbol ROW_i denotes the i-th row of upper-triangle part of the checked up adjacency matrix whereas the symbol row_i denotes the same row of transformed matrix $\boldsymbol{R}^{\mathrm{T}}\boldsymbol{A}\,\boldsymbol{R}$ (i.e. this row is composed of the entries $A_{r_i, r_{i+1}}, A_{r_i, r_{i+2}}, \ldots, A_{r_i, r_p}$). The rows 14 to 18 contain the very efficient accelerating method based on the existence of automorphisms in the verified graph. The vertices (indices) that are similar[29] with a vertex already used are removed from sets U_i, where the index i is bounded by $i < k$, where k is the minimum value of an index set for which $r_k \neq k$. Here it is very important to note that this approach changes the value of index i (determining the depth of searching tree) to a lower value, i.e. a huge part of the searching tree is pruned by this approach. Many branches which may give only the results already achieved are skipped.

In a difference from algorithm 1, the vertices are separately divided into subsets Y's for each level of depth-first searching tree. This is the main reason why these subsets

are denoted by two indices, $Y_k^{(i)}$; the lower and upper indices correspond to an index of vertex and to a level of searching tree, respectively.

Constructive Enumeration

The theory of canonical adjacency matrices (or canonically numbered graphs) elaborated in the previous section represents an effective tool for constructive enumeration of all possible canonical adjacency matrices from the prescribed family $\mathcal{F}_{pq}$. The basic idea of this method involves a recurrent construction of all $(s + 1)$-reduced canonical adjacency matrices from a given s-reduced canonical adjacency matrix. Repeating recurrently this process until all rows in upper-triangle part of adjacency matrices are filled by 1 and/or 0 entries we have to arrive at canonical adjacency matrices that are generated from the given s-reduced canonical adjacency matrix. Let $\boldsymbol{A}$ be an s-reduced canonical adjacency matrix, its form is [cf. Eq. (*12*)]

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{12}^{\mathrm{T}} & O \end{pmatrix}, \tag{20}$$

where the left-up corner submatrix A_{11} is of the type (s,s) whereas the submatrix A_{12} is of the type $(s, p - s)$. The submatrix A_{12} may be expressed by its column vectors as follows

$$A_{12} = (c_1, c_2, \ldots, c_{p-s}) . \tag{21}$$

The process of an extension of the s-reduced matrix $\boldsymbol{A}$ to an $(s + 1)$-reduced matrix

$$\boldsymbol{A}' = \begin{pmatrix} \boldsymbol{A}_{11}' & \boldsymbol{A}_{12}' \\ \boldsymbol{A}_{12}'^{\mathrm{T}} & \boldsymbol{O} \end{pmatrix} \tag{22a}$$

is uniquely determined by its submatrices

$$\boldsymbol{A}_{11}' = \begin{pmatrix} \boldsymbol{A}_{11} & \boldsymbol{c}_1 \\ \boldsymbol{c}_1^{\mathrm{T}} & 0 \end{pmatrix} \tag{22b}$$

$$\boldsymbol{A}_{12}' = \begin{pmatrix} \boldsymbol{c}_2 \, \boldsymbol{c}_3 \, \ldots \, \boldsymbol{c}_{p-s} \\ \alpha_2 \, \alpha_3 \, \ldots \, \alpha_{p-s} \end{pmatrix} = (c_2', c_3', \ldots, c_{p-s}') , \tag{22c}$$

where

$$c_i' = \begin{pmatrix} c_i \\ \alpha_i \end{pmatrix} \qquad (\text{for } i = 2, 3, \ldots, p - s) \tag{22d}$$

are new column vectors of $\boldsymbol{A}'_{12}$ formed from the column vectors of $\boldsymbol{A}_{12}$ by adding from bottom an integer $\alpha_i = 0$ or $\alpha_i = 1$. These added entries $\alpha_2, \alpha_3, \ldots, \alpha_{p-s}$ determine, in fact, whether the resulting extended $(s + 1)$-reduced adjacency matrix will be canonical or not. What we can do to ensure that the matrix $\boldsymbol{A}'$ will be at least potentially canonical? According to theorems 2 and 3 this matrix should be semicanonical, therefore the entries $\alpha_2, \alpha_3, \ldots, \alpha_{p-s}$ are selected in such a way that

$$c_2' \geq c_3' \geq \ldots \geq c_{p-s}' . \tag{23}$$

Unfortunately, the produced adjacency matrix $\boldsymbol{A}'$ is not automatically canonical, it is semicanonical and all its smaller reduced adjacency matrices are canonical. Therefore, we have to check whether a produced matrix $\boldsymbol{A}'$ is canonical or not. If it is not, then this matrix is rejected from our forthcoming considerations. These ideas are summarized by the following theorem.

Theorem 4. Canonical $(s + 1)$-reduced matrices $\boldsymbol{A}^{(1)}_{s+1}, \boldsymbol{A}^{(2)}_{s+1}, \ldots$ produced by an extension of the given canonical s-reduced adjacency matrix $\boldsymbol{A}_s$ represent all possible nonequivalent canonical $(s + 1)$-reduced adjacency matrices which have the initial matrix $\boldsymbol{A}_s$ as the s-reduced adjacency matrix.

In special cases we are able to say in advance that the adjacency matrix $\boldsymbol{A}'$ will be canonical or noncanonical. First, if all entries α_i from (*22c* – *22d*) are equal to zero, then the adjacency matrix $\boldsymbol{A}'$ is automatically canonical. This simple property of our extension process immediately follows from the fact that there cannot exist a permutation $R = (r_1, r_2, \ldots, r_p)$ represented by a permutation matrix $\boldsymbol{R}$ which would produce an equivalent adjacency matrix with code greater than that one of the original adjacency matrix $\boldsymbol{A}'$. In the opposite case the parental adjacency matrix $\boldsymbol{A}$ could not be canonical. Second, let $R = (r_1, r_2, \ldots, r_p)$ be a permutation represented by a permutation matrix $\boldsymbol{R}$, and let us assume that the codes of $\boldsymbol{A}$ and its equivalent counterpart are the same, $[\boldsymbol{A}] = [\boldsymbol{R}^T\boldsymbol{A}\boldsymbol{R}]$, i.e. $a_{ij} = a_{r_i r_j}$, for each $1 \leq i < j \leq p$. Now, let us assume that for a given pair of indices i and j we have $r_i = s + 1$ and $s + 1 < r_j \leq p$, then $a'_{r_i r_j} = 0$ if $a_{ij} = 0$. This property must be satisfied since in the opposite case we may find a permutation matrix $\boldsymbol{R}$ which will produce a code $[\boldsymbol{A}''] = [\boldsymbol{R}^T\boldsymbol{A}'\boldsymbol{R}]$, which would be greater than $[\boldsymbol{A}']$. Following this observation, when we are extending the s-reduced canonical adjacency matrix $\boldsymbol{A}$ to an $(s + 1)$-reduced adjacency matrix $\boldsymbol{A}'$, we have to set $\alpha_{r_j} = 0$ in the process of creation of the submatrix $\boldsymbol{A}'_{12}$ [see Eq. (*22c*)]. It means that we have a simple criterion of nullity of some entries α_i's from (*22c*) which may be simply embedded in the framework of canonicity check of the parental s-reduced adjacency matrix.

If we have to construct only the connected graphs, then there exists a simple criterion of connectivity of graphs that are semicanonically numbered. If a column of entries in the upper-triangle part of adjacency matrix is composed entirely of null entries, then a graph corresponding to this adjacency matrix should be disconnected. In this case the adjacency matrix may be expressed as a block sum of two submatrices that correspond to separate components of the graph.

Theorem 4 enables us to suggest a very simple and effective method for constructive enumeration of all possible nonequivalent adjacency matrix from the prescribed family $\mathcal{F}_{pq}$. The effectiveness of the method is based on the fact that only nonequivalent adjacency matrices are formed. According to theorem 4 entire construction of adjacency matrices from the family $\mathcal{F}_{pq}$ may be organized in a recurrent manner. We start from a 1-reduced adjacency matrix in which the submatrix $\boldsymbol{A}_{12}$ is a row vector composed of $(p - 1)$ entries. From this matrix we form all possible nonequivalent 2-reduced adjacency matrices by adding to the previous one a second row composed of $(p - 2)$ entries. In general, for the s-step, when all canonical $(s - 1)$ reduced adjacency matrices have been already constructed, we shall construct all their extensions, i.e. canonical s-reduced adjacency matrices. This simple recurrent procedure is stopped when s becomes equal to $p - 1$, from the produced adjacency matrices we select those, belonging to the prescribed family $\mathcal{F}_{pq}$. To reduce the total number of produced adjacency matrices we check in each step the excess of 1 entries in produced adjacency matrices, adjacency matrices with a greater number of 1 entries than the prescribed number q are skipped in the forthcoming step. The basic principles of this method are illustrated in Fig. 2.

Each level in this figure corresponds to a number of the rows filled in the adjacency matrix, e.g. the first level composed of two graphs corresponds to an adjacency matrix with the first row completely filled whereas other rows are still unfilled. In order to produce maximum codes of graphs a vertex numbered by 1 should be of greatest valence. Since we produce graphs with 5 vertices and 6 edges, the "greatest" valence is bounded from above (below) by 4 (3). Therefore in the first level we have only 2 starting structures. The second level is composed of graphs which have been produced by filling the second row in adjacency matrices, etc. We can see that the redundancy in production of graphs is minimal, the presented figure does not contain any blind branches, each was cut at the beginning.

Enumeration of Graphs with Prescribed Valences

The present method of constructive enumeration of graphs is very well suited when vertex valences of graphs to be constructed are prescribed. In addition to predicting some null entries in a row just created according to an automorphism of the current subgraph, other criteria could be applied for filling up the rows that are still empty.

Let us now construct graphs that are determined by a prescribed graphical[29] sequence of valence vertices. The approach can be best described by the following example.

Example 2. Suppose, that the graphs to be constructed are determined by a given graphical sequence of valences of vertices (333111), i.e. three vertices of the valence 3 and three vertices of the valence 1. Now, let us have two rows of connection matrix **A** already generated.

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & & & \\ 1 & 0 & & 0 & & \\ 0 & 1 & & & 0 & \\ 0 & 0 & & & & 0 \end{pmatrix} \longrightarrow \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

A **A'**

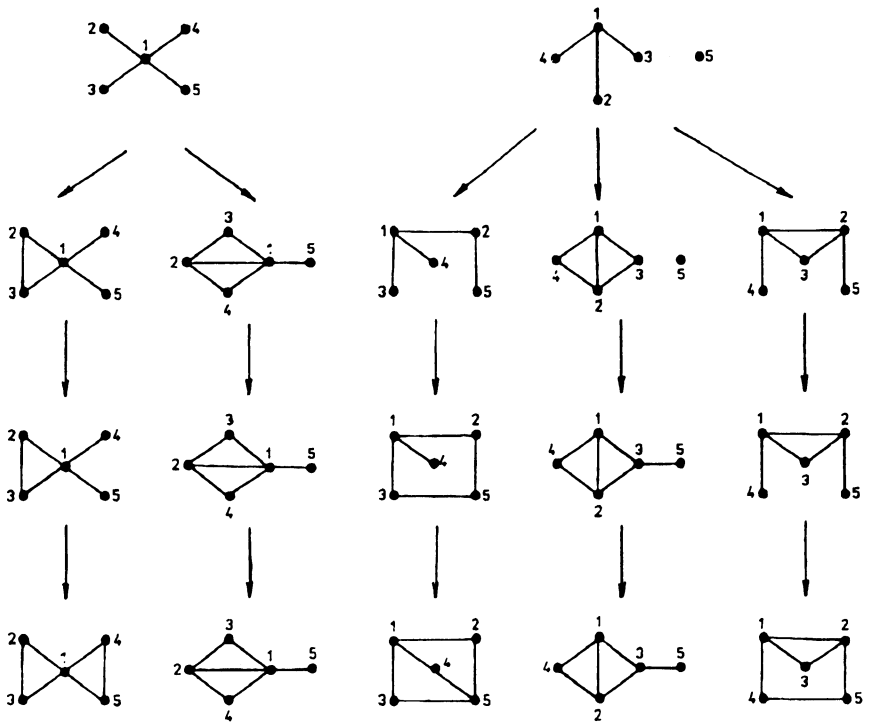

FIG. 2

An example of recurrent construction of all connected graphs composed of five vertices and six edges. All graphs are canonically numbered

TABLE I
Total numbers of connected nonisomorphic graphs with p vertices and q edges

q		p = 3	4	5	6	7	8	9	10
2	n^1	1							
	t^1	<0.1							
	n^2	1							
	t^2	<0.1							
3	n^1	1	2						
	t^1	<0.1	0.1						
	n^2	1	2						
	t^2	<0.1	0.1						
4	n^1		2	3					
	t^1		<0.1	0.1					
	n^2		2	3					
	t^2		<0.1						
5	n^1		1	5	6				
	t^1		<0.1	0.1	0.2				
	n^2		1	5	5				
	t^2		<0.1	0.1	0.1				
6	n^1		1	5	13	11			
	t^1		<0.1	0.1	0.2	0.2			
	n^2		1	5	12	9			
	t^2		<0.1	0.1	0.1	0.2			
7	n^1			4	19	33	23		
	t^1			0.1	0.2	0.5	0.6		
	n^2			4	17	29	18		
	t^2			0.1	0.2	0.4	0.4		
8	n^1			2	22	67	89	47	
	t^1			0.1	0.3	1.0	1.8	1.4	
	n^2			2	18	56	73	35	
	t^2			0.1	0.2	0.8	1.4	1.1	
9	n^1			1	20	107	236	240	106
	t^1			0.1	0.3	1.6	4.4	5.9	4.3
	n^2			1	14	79	182	185	75
	t^2			0.1	0.2	1.2	3.3	4.4	3.3
10	n^1			1	14	132	486	797	657
	t^1			0.1	0.3	2.1	8.4	16.8	19.5
	n^2			1	8	79	326	573	475
	t^2			0.1	0.1	1.3	5.5	11.8	14.1
11	n^1				9	138	814	2 075	2 678
	t^1				0.2	2.4	13.8	39.4	64.1
	n^2				3	59	430	1 278	1 792
	t^2				0.1	1.1	7.5	24.2	43.1

TABLE I
(*Continued*)

q		p							
		3	4	5	6	7	8	9	10
12	n^1				5	126	1 169	4 495	8 548
	t^1				0.1	2.4	20.0	80.7	178.3
	n^2				1	31	427	2 161	4 875
	t^2				<0.1	0.7	8.1	41.3	105.1
13	n^1				2	95	1 454	8 404	22 950
	t^1				0.1	2.1	26.0	146.2	438.2
	n^2					9	298	2 768	10 162
	t^2					0.2	6.8	58.0	213.3
14	n^1				1	64	1 579	13 855	53 863
	t^1				0.1	1.7	30.2	235.6	966.6
	n^2					2	134	2 616	16 461
	t^2					<0.1	3.7	63.5	364.7
15	n^1				1	40	1 515	20 303	112 618
	t^1				0.1	1.2	31.4	342.7	1 920.5
	n^2						35	1 714	20 346
	t^2						1.1	49.3	501.8
16	n^1					21	1 290	26 631	211 866
	t^1					0.7	30.1	456.4	3 473.2
	n^2						6	707	18 436
	t^2						0.1	24.8	523.2
17	n^1					10	970	31 400	361 342
	t^1					0.4	26.0	560.4	5 804.2
	n^2							154	11 477
	t^2							6.8	390.1
18	n^1					5	658	33 366	561 106
	t^1					0.2	20.7	629.8	8 984.1
	n^2							16	4 399
	t^2							0.5	185.8
19	n^1					2	400	31 996	795 630
	t^1					0.1	15.0	646.3	12 816.8
	n^2								845
	t^2								46.5
20	n^1					1	220	27 764	1 032 754
	t^1					0.1	10.0	611.0	16 822.2
	n^2								59
	t^2								3.9

n^1 number of graphs, valences of vertices are not restricted; n^2 number of graphs, valences of vertices are smaller or equal to 4; t^1, t^2 CPU times in seconds for PC AT compatible with 20-MHz clock.

As there has to be generated three rows for vertices with valence equal to one, and the first three rows correspond to vertices with higher valence, the rows for vertices with valence equal to one are only the last three ones. As the fourth and fifth row already has its "1" entry, the only remaining possibility for another "1" entry is on the sixth row and third column (or vice versa). Then the sequence of valences is obtained and all remaining empty entries can be filled by zeros. For generation of another matrix we can return to the second row. Generally, we can define for each row, which is not yet filled, a maximum number of "1" its entries, which it would achieve by filling all empty entries by "1". For this row we can define also a minimum number of 1-entries, which it would achieve by filling all empty entries by "0". These numbers for the set of rows can be compared with the required sequence of valences of vertices and according to that some rows can be filled immediately either by "1" entries or by "0" entries. For check of canonical numbering all the empty entries should be filled by zero. When none of permutation of required sequence of valences of vertices match the bounds of maximum and minimum valences given by already generated matrix, the current matrix is skipped[24].

RESULTS AND DISCUSSION

For simplicity the whole method is described for constructive enumeration of simple graphs, but after a slight modification it can be used for enumeration of multigraphs with vertices evaluated by atomic symbols – molecular graphs. Its main advantage is that it is not based on any a priori given heuristics or restrictions like a database of structures containing cycles, and in spite of that it is efficient. Faradzhev's concept of semicanonical adjacency matrices represents a very efficient theoretical tool helping to reduce considerably an enormous number of created graphs in which the canonicity of numbering has to be checked. Moreover, in the framework of the method simple criteria based on an existence of automorphisms in a graph just extended can be straightforwardly introduced. In Table I are presented results of our program written in C language for constructive enumeration of graphs with prescribed number of vertices and edges and/or with a prescribed sequence of vertex valences. All results given in this table were checked by an independent program[19]. We have constructively enumerated all connected graphs that are composed of from three to ten vertices and of at most twenty edges. What is very impressive in our numerical results, the program runs on an IBM PC-AT compatible (with 20-MHz clock) and generates 25 – 50 graphs (with ten vertices) per second.

REFERENCES

1. Lederberg J.: *Computation of Molecular Formulas for Mass Spectroscopy*. Holden–Day, San Francisco 1964.
2. Lederberg J., Sutherland G. L., Buchanan B. G., Feigenbaum E. A., Robertson A. V., Duffield A. M., Djerassi C.: J. Am. Chem. Soc. *21*, 2973 (1969).
3. Read R. C. in: *Chemical Application of Graph Theory* (A. T. Balaban, Ed.), p. 25. Academic Press, London 1976.
4. Kvasnička V., Pospíchal J.: Collect. Czech. Chem. Commun. *56*, 1777 (1991).
5. Heap B. R.: *Graph Theory and Computing*, p. 47. Academic Press, New York 1972.
6. Baker H. H., Dewdney A. K., Szilard A. L.: Math. Comput. *127*, 833 (1974).
7. Bussemaker F. S., Cobejlic S., Cvetkovic L. M., Seidel J. J.: *Computing Investigation of Cubic Graphs*. Technical Report 76-WSK-01, Technical University Eindhowen 1976.
8. Masinter L. M., Sridharan N. S., Lederberg J., Smith D. H.: J. Am. Chem. Soc. *96*, 7702 (1974).
9. Smith D. H., Masinter M. L., Sridharan N. S. in: *Computer Representation and Manipulation of Chemical Information* (W. T. Wipke, S. R. Heller, R. J. Feldmann and E. Hyde, Eds), p. 287. Wiley, New York 1974.
10. Lindsay R. K., Buchanan B. G., Feigenbaum E. A., Lederberg J.: *Applications of Artificial Intelligence for Organic Chemistry: The DENDRAL Project*. McGraw–Hill, New York 1980.
11. Balaban A. T. in: *Chemical Application of Graph Theory* (A. T. Balaban, Ed.), p. 63. Academic Press, London 1976.
12. Bauer J.: Tetrahedron Comput. Methodology *2*, 269 (1989).
13. Razinger M., Župan J., Novic M.: Microchim. Acta *111* (1986).
14. Bangov I. P.: J. Chem. Inf. Comput. Sci. *30*, 277 (1990).
15. Read R. C., Corneil D. G.: J. Graph Theory *1*, 339 1977).
16. Laue R.: Bayreuther Math. Schriften *28*, 111 (1989).
17. Kerber A., Laue R., Moser D.: Anal. Chim. Acta *235*, 221 (1990).
18. Hager R., Kerber A., Laue R., Moser D., Weber W.: Bayreuther Math. Schriften *31*, 19 (1990).
19. Kvasnička V., Pospíchal J.: J. Chem. Inf. Comput. Sci. *30*, 99 (1990).
20. Kvasnička V., Pospíchal J.: Chemometrics Intell. Lab. Syst. *11*, 137 (1991).
21. Hendrickson J. B., Parks C. A.: J. Chem. Inf. Comput. Sci. *31*, 101 (1991).
22. Morgan H. L.: J. Chem. Doc. *5*, 107 (1965).
23. Hendrickson J. B., Toczko A. G.: J. Chem. Inf. Comput. Sci. *23*, 171 (1983).
24. Faradzhev I. A. in: *Algoritmicheskie issledovaniya v kombinatorike* (I. A. Faradzhev, Ed.), p. 11. Nauka, Moscow 1978.
25. Faradzhev I. A. in: *Algoritmicheskie issledovaniya v kombinatorike* (I. A. Faradzhev, Ed.), p. 3. Nauka, Moscow 1978. Faradzhev I. A.: Lecture, Orsay 9th July 1976. Colloques internationaux C.N.R.S. No 260 – Problemes combinatoires et theorie des graphes. Paris 1978.
26. Arlazarov V. L., Zuev I. I., Uskov A. V., Faradzhev I. A.: Zh. Vychisl. Mat. Mat. Fiz. *14*, 737 (1974).
27. Zaidenko V. A., Ivanov A. V., Rozenfeld M. Z., Faradzhev I. A. in: *Algoritmicheskie issledovaniya v kombinatorike* (I. A. Faradzhev, Ed.) p. 19. Nauka, Moscow 1978.
28. Molodtsov S. G., Piottuch-Peletsky V. N.: Vychisl. Sist. *103*, 51 (1984).
29. Harary F.: *Graph Theory*. Addison Wesley, Reading 1969.
30. Mac Lane S., Birkhoff G.: *Algebra*. The Macmillan, New York 1974.
31. Lawler E. L., Wood D. E.: J. Op. Res. Soc. Am. *14*, 217 (1966).
32. Randic M.: J. Chem. Inf. Comput. Sci. *17*, 171 (1977).
33. Brandt J., Bauer J., Frank R. M., von Scholey A.: Chem. Scr. 18, 53 (1981).

Translated by the author (J. P.).